Winter 2008

CS 142-01: Computer Programming - II

Michael Ondrasek
Wright State University - Main Campus, michael.ondrasek@wright.edu

CS 142 Computer Programming - II

Winter 2008 - Lecture (22099)

Last Update Monday Decemberr 10, 2007 at 10:00 a.m.

Monday and Wednesday 2:45 – 4:00 p.m., Oelman 306
Plus **one** of the following lab sections:

Sect	Time	Day	Room
22100 5	1:30 p.m. - 2:20 p.m.	W	RC 346
22101 6	4:10 p.m. - 5:00 p.m.	W	RC 346

Description: The concepts introduced in CS 141 are developed in greater detail and depth with the Java programming language. Topics include object oriented programming, graphics, development of user interfaces and handling runtime errors with an emphasis on program verification and testing. Students must register for both lecture and one laboratory section. 4 credit hours. Prerequisite: CS 141 (Computer Programming I) and MTH 127 (College Algebra) or equivalent.

Instructor: Mr. Michael Ondrasek (Office RC 160 inside the student study area RC 158) email is michael.ondrasek@wright.edu. Office hours: by appointment.

Teaching Assistants: Mr. Allan Rwabutaza, rwabutaza.2@wright.edu and Ms. Lisa Jacob, jacob.13@wright.edu. Their office location is RC 316; office hours to be announced.

Textbook: Starting Out with Java 5, From Control Structures through Objects, 3rd edition by Tony Gaddis, Pearson Addison Wesley, 2008, ISBN 0-321-47927-0.

Textbook Web Resources: The **text CD** and ftp://ftp.aw.com/cseng/authors/gaddis/java/Control_Objects/3e. for the student support page. On the student support web page, at the text listing, there are links to download files containing the text appendices, and sample labs/programs. Other useful links include the location of the Sun JDK.

WebCT: http://wisdom.wright.edu If you are new to WebCT, please read the opening web page instructions for students. WebCT allows you access to your grades as well as lab assignments and submittals. We will post much of the course materials on WebCT.

Grading: Mid-term exam and quizzes: 20% (see calculation below); comprehensive final: 30%; programming lab assignments: 50%.

Mid-term and quizzes = 100*(YourMidTermScore+YourQuizPoints)/(MaxMidTermPoints + 0.85*MaxQuizPoints)

Final grade is based on the course average: **A:** 100-90, **B**: less than 90-80, **C:** less than 80-70, **D:** less than 70-60, **F:** less than 60-0 **unless you get less than 70% of the possible points on your programming lab assignments in which case you fail the entire course regardless of your overall course average.**

Policy: There are no late/early/makeup exams or quizzes unless verifiable emergency and acceptable documentation in writing is provided to the Instructor. Although verbal or e-mail notification can be provided, written documentation is required. Quizzes may be unannounced and/or take-home. Also some quizzes may have zero point value (practice). No quiz scores will be dropped, but only portion of the points will be counted as noted in the grade calculation. **All work must be your own; sharing of program code/take-home quiz solutions will result in a**

grade of "zero" for all those involved. Official university policy will be followed in cases of academic dishonesty. Do not show others your programs and do not look at someone else's code. However, sharing ideas and general computer skills with others outside of class is encouraged.

Expectations of Students: Attendance will be taken and recorded; but attendance at lecture is not required although it is strongly encouraged and expected. The Instructor considers it essential to your success in this course that you attend all lectures and lab sessions. Students are expected to study the text. **Even when you don't attend class, you are still responsible for material covered in lecture, lab, and in your text readings.** If you miss a lecture, you may also miss a quiz. If you miss an unexcused quiz you will receive a zero score. Students are expected to be on time for lecture and lab sessions: lectures and labs start promptly. Early departure from lecture or lab may be unavoidable, but it is expected that this would be quite unusual. The Instructor feels that it is important that you have your own copy of the correct textbook and edition indicated above. If you have a computer at home, it is important that you practice programming using software discussed in class. If you do not have a computer, it is expected that you will use the computers in Russ Center Room 152C (or other campus locations) to practice programming skills. Questions are encouraged in lecture and lab; however, if there are no questions it is assumed that students understand the lecture, have read, and understand the text and lab materials. If you are having trouble with programs or text readings, it is expected that you will ask questions in class, come during office hours for help, or make an appointment to discuss your questions as needed. Corresponding with the Instructor or Teaching Assistants by e-mail is a good way to get help with text readings or programming assignments. Finally, it is expected that students will follow the Instructor's recommendations concerning printing of slides and other course materials. In order to minimize handouts, you are expected to print your own copies before lecture whenever possible. The computers in Russ Center Room 152C provide all registered students with the ability to freely print their own copies. Please follow guidelines given in lecture on how to make the best use of the computing and printing resources.

Suggestions: Get an early start on each programming assignment. Most often you will not complete the programming assignment in the 50-minute lab sessions. You are urged to budget your lab time wisely and expect to spend additional time outside of the formal lab to complete your programming assignments. Get acquainted with the CD/online textbook materials: study the text Preface carefully. You should print, review, and study online materials recommended by the Instructor and Teaching Assistants. You can download the source code for the text examples to try them out. Whenever possible study your text in front of a computer and actively get involved in trying out the programming concepts on your own. You should try to do all text checkpoint, review questions, and exercises. This can be the most effective way to be successful in the course. If you are uncertain about how you should do this, please discuss with the Instructor or Teaching Assistant. It would be a very good idea to get your own USB 2.0 compatible flash drive (also known as a "thumb drive" or "min-drive") for use in labs and possibly at home. See the Instructor or Teaching Assistants for recommendations and usage. **Always backup your programs!** Keep copies of your work in several different places. E-mail yourself a backup copy.

Programs: Programming lab assignments will be issued in class, during the lab sessions, or on WebCT. Each assignment will state the due date. Assignments usually will be one or possibly two weeks in duration. As noted above, you must earn at least **70%** of the possible points on lab assignments in order to pass the course. Programming assignments are to be submitted on the due date. Late assignments are accepted at the discretion of the Teaching Assistant, who will impose a late penalty.

Syllabus Changes: The Instructor will not make changes to this syllabus without notification and understanding of all the students in the class. New paper copies will be provided. Changes would be required for the following reasons: (1) to correct mistakes, (2) to improve student learning, (3) to clarify misunderstands, or (4) to correct serious inconsistencies in policies and/or content compared to other concurrent lecture sections sharing the same labs.

Schedule: See the table below. Topics and order of topics may vary. Exam dates are firm. The topics to be covered each week are listed, followed by the accompanying sections in the text. Not all sections listed are directly covered in detail in class. This schedule is subject to change.

Week	Topic	Readings*
1	Introduction to Objects and Classes, Constructors, Overloading Methods and Constructors, Scope of Instance Fields, Packages and *import* Statements	Chapter 6
2	Creating Windows, GUI classes with a main Method, Layout Managers, Radio Buttons, Borders, Splash Screens, debugging	Chapter 7
3	Static Class Members, Object Passing to and Returning from methods, toString and equals methods	Chapter 9.1 – 9.5
4	Aggregation, the *this* key word, Enumerated Types, Garbage Collection, Class Design, Errors to Avoid	Chapter 9.6 – 9.12
5	Wrapper Classes, Character Class, StringBuilder Class **Exam: Wednesday February 6th**	Chapter 10.1 – 10.4
6	Tokenizing Strings, Numeric Data Wrapper Classes, Errors to Avoid	Chapter 10.5 – 10.8
7	Inheritance, Superclass Constructors/Methods, Chains of Inheritance, Polymorphism, Abstract Classes and Methods, Interfaces	Chapter 11
8	Handling and Throwing Exceptions, Binary Files, Random Access Files, Object Serialization	Chapter 12
9	Applets, HTML, Drawing Shapes, Handling Mouse Events, Timer Objects, Playing Audio. Course Evaluation	Chapter 14
10	Swing, AWT, Lists, Combo Boxes, Labels, Buttons, Mnemonics, Tool Tips, File and Color Choosers, Menus, Text Components, Sliders. Review for Final Exam	Chapter 13
Finals Week	**Comprehensive Final Exam: Wednesday March 19th (3:15 – 5:15 pm)**	

*Whenever any Chapter is assigned, you should also study the corresponding Review Questions and Exercises found at the end of each chapter.